Searchmarkets.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="SearchMarkets.aspx.cs" Inherits="Default3" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Search markets</title>

<style type="text/css">

header {

background-color:black;

color:white;

text-align:center;

padding:5px;

}

#tb1 {

font-family: verdana,arial,sans-serif;

font-size:13px;

color:#333333;

border-color: #a9c6c9;

border-collapse: collapse;

}

#tb1 th {

padding: 8px;

border-style: solid;

border-color: #a9c6c9;

}

#tb1 td {

padding: 8px;

background: #ebecda;

background: url(data:image/svg+xml;base64,);

background: -moz-linear-gradient(top, #ebecda 0%, #e0e0c6 40%, #ceceb7 100%);

background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#ebecda), color-stop(40%,#e0e0c6), color-stop(100%,#ceceb7));

background: -webkit-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: -o-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: -ms-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: linear-gradient(to bottom, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

border: 1px solid #999999;

}

nav {

line-height:30px;

background-color:#9C8888;

height:440px;

width:275px;

float:left;

padding:5px;

}

section {

width:550px;

float: left;

padding:10px;

}

footer

{

background-color:black;

color:white;

clear:both;

text-align:center;

padding: 10px;

position:absolute;

bottom:0;

width:100%;

}

#outputdiv

{

position:fixed;

overflow: auto;

width: auto;

height: auto;

background-color: #0156c0;

color: white;

FONT-SIZE: .8em;

margin: 0px 0px 0px 0px;

padding: 5px;

}

</style>

</head>

<body>

<header>

<h1>Search Markets By Place</h1>

</header>

<nav>

<br />

<br />

<h3 align="center">Search markets by place</h3>

<p>This service provides the list of markets in a place entered by user.The user must enter a query as Markets in California.</p>

</nav>

<section>

<form id="form1" runat="server">

<div id="outputdiv" runat="server">

<input type="text" id="market" size="70" placeholder="Enter query(ex. markets in newyork)" runat="server"/>

<input type="button" value="submit" runat="server" onserverclick="searchbyplace"/>

<label id="d1" runat="server"></label>

<table id="tbl"

style="border-width:1; border-color:Black" border="2"

runat="server"></table>

</div>

</form>

</section>

<footer>

Copyright © EMercado.com

</footer>

</body>

</html>

Searchmarkets.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.IO;

using System.Net;

using System.Text;

using System.Xml;

using System.Web.UI.HtmlControls;

public partial class Default3 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

public void searchbyplace(object sender, EventArgs e)

{

String URL = "https://maps.googleapis.com/maps/api/place/textsearch/xml?query=";

String key = "&key=AIzaSyCIDxyi3jR-1Reb8xaqLKzXBD5-UQ6Ys64";

String res = SearchMarkets(market.Value, URL,key);

tbl.Controls.Clear();

/\*

for (int i = 0; i < 10; i++)

{

TableRow rowNew = new TableRow();

tbl.Controls.Add(rowNew);

for (int j = 0; j < 3; j++)

{

TableCell cellNew = new TableCell();

cellNew.Controls.Add(lblNew);

cellNew.Controls.Add(imgNew);

rowNew.Controls.Add(cellNew);

}

}\*/

String xmlUrl = "E:\\softwares\\EMercado\\App\_Data\\a.xml";

// XmlTextReader reader = new XmlTextReader(xmlUrl);

/\* while (reader.Read())

{

switch (reader.NodeType)

{

case XmlNodeType.Element: // The node is an element.

Response.Write("<" + reader.Name + ">");

break;

case XmlNodeType.Text: //Display the text in each element.

Response.Write(reader.Value + "<br />");

break;

case XmlNodeType.EndElement: //Display the end of the element.

Response.Write("</" + reader.Name + ">");

break;

}

}

reader.Close();\*/

//

//bharat

XmlDocument doc = new XmlDocument();

doc.Load(xmlUrl);

XmlNode PlaceSearchNode =

doc.SelectSingleNode("/PlaceSearchResponse");

XmlNodeList MarketNodeList =

PlaceSearchNode.SelectNodes("result");

HtmlTableRow row1 = new HtmlTableRow();

HtmlTableCell headerCell = new HtmlTableCell();

HtmlTableCell headerCell1 = new HtmlTableCell();

headerCell.InnerHtml = "<b> Market Name </b>";

headerCell1.InnerHtml = "<b> Address </b>";

row1.Cells.Add(headerCell);

row1.Cells.Add(headerCell1);

row1.BorderColor = "blue";

tbl.Rows.Add(row1);

foreach (XmlNode node in MarketNodeList)

{

HtmlTableRow row = new HtmlTableRow();

HtmlTableCell nameCell = new HtmlTableCell();

HtmlTableCell addCell = new HtmlTableCell();

String s;

s = node.SelectSingleNode("name").InnerText;

nameCell.InnerText = s;

s = node.SelectSingleNode("formatted\_address").InnerText;

addCell.InnerText = s;

row.Cells.Add(nameCell);

row.Cells.Add(addCell);

row.BorderColor = "blue";

tbl.Rows.Add(row);

}

File.Delete(xmlUrl);

}

public String SearchMarkets(String queryString, String Url, String Key)

{

String targetUrl = Url+queryString+Key;

WebRequest request = WebRequest.Create(targetUrl);

request.Method = "GET";

WebResponse response = request.GetResponse();

StreamReader sr = new StreamReader(response.GetResponseStream());

String resp = sr.ReadToEnd();

String path = "E:\\softwares\\EMercado\\App\_Data\\a.xml";

StreamWriter sw = File.CreateText(path);

sw.Write(resp);

sw.Close();

return resp;

}

}

Product search\_price.htm

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Category and Best Price</title>

<script src="http://code.jquery.com/jquery-1.7.js"

type="text/javascript"></script>

<script type="text/javascript">

function getData() {

var type = document.getElementById("type");

var category = type.options[type.selectedIndex].value;

var url = "http://kc-sce-cs551-2.kc.umkc.edu/aspnet\_client/Group12/Project/CategorySearch/CategorySearch/Service1.svc/SearchProductsByCategory/"

var targetUrl = url + category;

// debugger

//alert(targetUrl);

$.get(targetUrl, function (data) {

var table = document.getElementById("categoryData");

var count = table.rows.length;

// alert(count);

/\*for (i = 0; i < count; i++)

table.deleteRow(i);

debugger

\*/

// alert(data);

//alert(data[0]);

var data1 = data;

var j = 1;

var header = table.createTHead();

var row = header.insertRow(0);

var cell = row.insertCell(0);

cell.innerHTML = "<b> Items Available in " + category + "</b>";

var cell1 = row.insertCell(1);

cell1.innerHTML = "<b>Best Price</b>";

var cell2 = row.insertCell(2);

cell2.innerHTML = "<b>Average Price </b>";

if (data.length == 0) {

var header = table.createTHead();

var row = header.insertRow(1);

var cell = row.insertCell(0);

cell.innerHTML = " No Items Available in this " + category;

} else {

for (i = 0; i < data.length; i++) {

var row = table.insertRow(j++);

var cell1 = row.insertCell(0);

var cell2 = row.insertCell(1);

var cell3 = row.insertCell(2);

cell1.innerHTML = data[i];

var price = getBestPrice(data[i], cell2, cell3);

// alert(price);

// cell2.innerHTML = price;

}

}

});

}

function getBestPrice(data, cell2, cell3) {

var priceData = "";

var Url = "http://kc-sce-cs551-2.kc.umkc.edu/aspnet\_client/Group12/Project/BestPrice/BestPrice/Service1.svc/GetBestPrice/";

var item = data;

var targetUrl = Url + item;

// alert(targetUrl);

$.get(targetUrl, function (dataa) {

priceData = dataa;

if (priceData == "") {

cell2.innerHTML = "";

cell3.innerHTML = "";

} else {

// alert(dataa);

// alert(priceData);

var p1 = priceData.split(" ");

cell2.innerHTML = p1[0];

cell3.innerHTML = p1[1];

}

});

// debugger

// return priceData;

}

</script>

<style type="text/css">

header {

background-color:black;

color:white;

text-align:center;

padding:5px;

}

nav {

line-height:30px;

background-color:#9C8888;

height:440px;

width:275px;

float:left;

padding:5px;

}

section {

width:550px;

float: left;

padding:10px;

}

table.maintable {

font-family: verdana,arial,sans-serif;

font-size:13px;

color:#333333;

border-color: #a9c6c9;

border-collapse: collapse;

}

table.maintable th {

padding: 8px;

border-style: solid;

border-color: #a9c6c9;

}

table.maintable td {

padding: 8px;

background: #ebecda;

background: url(data:image/svg+xml;base64,);

background: -moz-linear-gradient(top, #ebecda 0%, #e0e0c6 40%, #ceceb7 100%);

background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#ebecda), color-stop(40%,#e0e0c6), color-stop(100%,#ceceb7));

background: -webkit-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: -o-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: -ms-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: linear-gradient(to bottom, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

border: 1px solid #999999;

}

.oddrowcolor{

background-color:#d4e3e5;

}

.evenrowcolor{

background-color:#c3dde0;

}

#type{

font-family: verdana,arial,sans-serif;

font-size:13px;

color:#333333;

border-color: #a9c6c9;

border-collapse: collapse;

}

table.categoryData

{

font-family: verdana,arial,sans-serif;

font-size:13px;

color:#333333;

border-color: #a9c6c9;

border-collapse: collapse;

}

table.categoryData th {

padding: 5px;

border-style: solid;

border-color: #a9c6c9;

}

table.categoryData td {

padding: 5px;

background: #ebecda;

background: url(data:image/svg+xml;base64,);

background: -moz-linear-gradient(top, #ebecda 0%, #e0e0c6 40%, #ceceb7 100%);

background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#ebecda), color-stop(40%,#e0e0c6), color-stop(100%,#ceceb7));

background: -webkit-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: -o-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: -ms-linear-gradient(top, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

background: linear-gradient(to bottom, #ebecda 0%,#e0e0c6 40%,#ceceb7 100%);

border: 1px solid #999999;

}

#categorydiv

{

position:fixed;

overflow: auto;

width:540px;

padding:5px;

height:440px;

background-color:silver;

}

footer

{

background-color:black;

color:white;

clear:both;

text-align:center;

padding: 10px;

position:absolute;

bottom:0;

width:100%;

}

#infobar

{

margin-left:850px;

background-color:#9C8888;

padding:5px;

height:440px;

font-family: verdana,arial,sans-serif;

font-size:16px;

color:#333333;

border-color: #a9c6c9;

border-collapse: collapse;

}

}

</style>

</head>

<body>

<header>

<h1>Category Based Product Search</h1>

</header>

<nav>

<div id="maindiv">

<br>

<table border="2" align="center" class="maintable" id="maincolor">

<br><br><tr>

<td>

Select Category To List Products Available

</td>

</tr>

<tr align="center">

<td><br>

<select id="type" class="selectpicker" multiple>

<option value="Root Vegetables">Root Vegetables</option>

<option value="Leafy Vegetables">Leafy Vegetables</option>

<option value="Fruits">Fruits</option>

<option value="Flower Buds">Flower Buds</option>

<option value="Stem Vegetables">Stem Vegetables</option>

<option value="Podded Vegetables">Podded Vegetables</option>

</select>

<br><br></td>

</tr>

<tr align="center">

<td>

<button onclick="getData()" class="get">Submit</button>

</td>

</tr>

</table>

</div>

</nav>

<section>

<div id="categorydiv" >

<table id="categoryData" class="categoryData" border="2" align="center">

</table>

</div>

</section>

<div id="infobar">

<br /><br /> <br /><br /> <p align="center">This page displays the products available based on the category choosed by the user.It enables customers with ease to choose the kind of products they need.</p>

</div>

<footer>

Copyright © EMercado.com

</footer>

</body>

</html>

Farmerreview.htm

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function () {

$("#design1").hide();

$("button.design1").click(function () {

$("#design1").toggle();

$("#image").toggle();

});

});

$(document).ready(function () {

$("#design1").hide();

$("button.design2").click(function () {

$("#design1").hide();

$("#design2").toggle();

$("#image").toggle();

});

});

function insertData() {

var url = "http://kc-sce-cs551-2.kc.umkc.edu/aspnet\_client/Group12/Project/FarmerReview/FarmerReview/Service1.svc/PutFarmerReview/design1/";

var farmerName = document.getElementById("fname").value;

var ratings = document.getElementById("rating").value;

var feedback = document.getElementById("fb").value;

var targetURL = url + farmerName + "/" + ratings+ "/" + feedback;

$.get(targetURL, function (data) {

var report = "Success";

if (data == report) {

alert("Your Data Successfully Inserted");

}

});

}

</script>

<style>

.design1

{

background-color: #9F5151;

-moz-border-radius: 5px;

-webkit-border-radius: 5px;

border-radius:6px;

color: #fff;

font-family: 'Oswald';

font-size: 20px;

text-decoration: none;

cursor: pointer;

border:none;

}

.design1:hover {

border: none;

background:#704848;

box-shadow: 0px 0px 1px #777;

}

.design2

{

background-color: #9F5151;

-moz-border-radius: 5px;

-webkit-border-radius: 5px;

border-radius:6px;

color: #fff;

font-family: 'Oswald';

font-size: 20px;

text-decoration: none;

cursor: pointer;

border:none;

}

.design2:hover {

border: none;

background:#704848;

box-shadow: 0px 0px 1px #777;

}

table.tab

{

position:relative;

left: 50%;

margin:0px;padding:0px;

width:100%;

box-shadow: 10px 10px 5px #888888;

border:1px solid #000000;

-moz-border-radius-bottomleft:9px;

-webkit-border-bottom-left-radius:9px;

border-bottom-left-radius:9px;

-moz-border-radius-bottomright:9px;

-webkit-border-bottom-right-radius:9px;

border-bottom-right-radius:9px;

-moz-border-radius-topright:9px;

-webkit-border-top-right-radius:9px;

border-top-right-radius:9px;

-moz-border-radius-topleft:9px;

-webkit-border-top-left-radius:9px;

border-top-left-radius:9px;

}.tab table{

width:100%;

height:100%;

margin:0px;padding:0px;

}.tab:last-child td:last-child {

-moz-border-radius-bottomright:9px;

-webkit-border-bottom-right-radius:9px;

border-bottom-right-radius:9px;

}.tab tr:first-child td:first-child {

-moz-border-radius-topleft:9px;

-webkit-border-top-left-radius:9px;

border-top-left-radius:9px;

}.tab tr:first-child td:last-child {

-moz-border-radius-topright:9px;

-webkit-border-top-right-radius:9px;

border-top-right-radius:9px;

}.tab tr:last-child td:first-child{

-moz-border-radius-bottomleft:9px;

-webkit-border-bottom-left-radius:9px;

border-bottom-left-radius:9px;

}.tab tr:hover td{

background-color:#82c0ff;

background:-o-linear-gradient(bottom, #82c0ff 5%, #56aaff 100%); background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05, #82c0ff), color-stop(1, #56aaff) );

background:-moz-linear-gradient( center top, #82c0ff 5%, #56aaff 100% );

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#82c0ff", endColorstr="#56aaff"); background: -o-linear-gradient(top,#82c0ff,56aaff);

}.tab tr:first-child td{

background:-o-linear-gradient(bottom, #0069d3 5%, #007fff 100%); background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #0069d3), color-stop(1, #007fff) );

background:-moz-linear-gradient( center top, #0069d3 5%, #007fff 100% );

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#0069d3", endColorstr="#007fff"); background: -o-linear-gradient(top,#0069d3,007fff);

background-color:#0069d3;

border:0px solid #000000;

text-align:center;

border-width:0px 0px 1px 1px;

font-size:18px;

font-family:Comic Sans MS;

font-weight:bold;

color:#ffffff;

}.tab tr:first-child:hover td{

background:-o-linear-gradient(bottom, #0069d3 5%, #007fff 100%); background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #0069d3), color-stop(1, #007fff) );

background:-moz-linear-gradient( center top, #0069d3 5%, #007fff 100% );

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#0069d3", endColorstr="#007fff"); background: -o-linear-gradient(top,#0069d3,007fff);

background-color:#0069d3;

}.tab tr:first-child td:first-child{

border-width:0px 0px 1px 0px;

}.tab tr:first-child td:last-child{

border-width:0px 0px 1px 1px;

}.tab td{

background:-o-linear-gradient(bottom, #56aaff 5%, #82c0ff 100%); background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #56aaff), color-stop(1, #82c0ff) );

background:-moz-linear-gradient( center top, #56aaff 5%, #82c0ff 100% );

filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#56aaff", endColorstr="#82c0ff"); background: -o-linear-gradient(top,#56aaff,82c0ff);

background-color:#56aaff;

border:1px solid #000000;

border-width:0px 1px 1px 0px;

text-align:left;

padding:7px;

font-size:12px;

font-family:Comic Sans MS;

font-weight:bold;

color:#000000;

}.tab tr:last-child td{

border-width:0px 1px 0px 0px;

}.tab tr td:last-child{

border-width:0px 0px 1px 0px;

}.tab tr:last-child td:last-child{

border-width:0px 0px 0px 0px;

}

header {

background-color:black;

color:white;

text-align:center;

padding:5px;

}

nav {

line-height:30px;

background-color:#9C8888;

height:300px;

width:200px;

float:left;

padding:5px;

}

section {

width:350px;

float: left;

padding:10px;

}

footer

{

background-color:black;

color:white;

clear:both;

text-align:center;

padding: 10px;

}

#image

{

position: relative;

top: 45px;

left: -40px;

}

</style>

</head>

<body>

<header>

<h1>Feedback Page</h1>

</header>

<nav>

<button class="design1">Add Farmer Review</button><br><br>

<button class="design2">Show Farmer Review</button><br>

</nav>

<img src="reviews.png" style="width:304px;height:228px" id="image"/>

<section>

<div id="design1"><p>

<table border="5" width="30%" class="tab" style="left:80%">

<tr>

<td>

<h1>Add Farmer Review</h1>

</td>

</tr>

<tr>

<td>

<h3>Farmer Name: <input type="text", id="fname"><br><br>

</td>

</tr>

<tr>

<td>

Ratings(1 to 5): <input type="number", id="rating", min="1", max="5"><br><br>

</td>

</tr>

<tr>

<td>

Feedback: <textarea rows="5", cols="50", id="fb"></textarea>

</td>

</tr>

<tr>

<td>

<button id="submit" onclick="insertData()">Submit</button>

</td>

</tr>

</table>

</h3>

</p></div>

</table>

</div>

</section>

<footer>

Copyright © EMercado.com

</footer>

</body>

</html>

**Services:**

**Bestprice**

**Iservice1.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.ServiceModel.Web;

using System.Text;

namespace BestPrice

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both code and config file together.

[ServiceContract]

public interface IService1

{

[OperationContract]

string GetData(int value);

[OperationContract]

String BestPrice(String itemName);

[OperationContract]

CompositeType GetDataUsingDataContract(CompositeType composite);

// TODO: Add your service operations here

}

// Use a data contract as illustrated in the sample below to add composite types to service operations.

[DataContract]

public class CompositeType

{

bool boolValue = true;

string stringValue = "Hello ";

[DataMember]

public bool BoolValue

{

get { return boolValue; }

set { boolValue = value; }

}

[DataMember]

public string StringValue

{

get { return stringValue; }

set { stringValue = value; }

}

}

}

Service.svc.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.ServiceModel.Web;

using System.Text;

using System.Data.SqlClient;

using System.Configuration;

namespace BestPrice

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.

public class Service1 : IService1

{

public string GetData(int value)

{

return string.Format("You entered: {0}", value);

}

public CompositeType GetDataUsingDataContract(CompositeType composite)

{

if (composite == null)

{

throw new ArgumentNullException("composite");

}

if (composite.BoolValue)

{

composite.StringValue += "Suffix";

}

return composite;

}

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "GetBestPrice/{itemName}")]

public String BestPrice(String itemName)

{

String bestPrice = "";

String avgPrice = "";

//Declare Connection by passing the connection string from the web config file

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["MercadoDb"].ConnectionString);

//Open the connection

conn.Open();

//Declare the sql command

SqlCommand getPriceCmd = new SqlCommand("Select min(Price),avg(Price) from FarmerItems where itemName ='" + itemName + "'",conn);

SqlDataReader reader = getPriceCmd.ExecuteReader();

while (reader.Read())

{

bestPrice = reader[0].ToString();

avgPrice = reader[1].ToString();

}

reader.Close();

//close the connection

conn.Close();

return bestPrice+" " +avgPrice;

}

}

}

Webconfig

<?xml version="1.0"?>

<configuration>

<connectionStrings>

<add name="MercadoDb" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=MercadoDatabase;Integrated Security=SSPI" providerName="System.Data.SqlClient"/>

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.0" />

</system.web>

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior>

<!-- To avoid disclosing metadata information, set the value below to false and remove the metadata endpoint above before deployment -->

<serviceMetadata httpGetEnabled="true"/>

<!-- To receive exception details in faults for debugging purposes, set the value below to true. Set to false before deployment to avoid disclosing exception information -->

<serviceDebug includeExceptionDetailInFaults="false"/>

</behavior>

</serviceBehaviors>

</behaviors>

<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />

</system.serviceModel>

<system.webServer>

<modules runAllManagedModulesForAllRequests="true"/>

</system.webServer>

</configuration>

Category search Service

Iservice1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.ServiceModel.Web;

using System.Text;

namespace CategorySearch

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both code and config file together.

[ServiceContract]

public interface IService1

{

[OperationContract]

String[] SearchProductsByCategory(String categoryName);

}

}

Service1.svc.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.ServiceModel.Web;

using System.Text;

using System.Data.SqlClient;

using System.Configuration;

namespace CategorySearch

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.

public class Service1 : IService1

{

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "SearchProductsByCategory/{categoryName}")]

public String[] SearchProductsByCategory(String categoryName)

{

String[] productNames;

//Declare Connection by passing the connection string from the web config file

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["MercadoDb"].ConnectionString);

//Open the connection

conn.Open();

SqlCommand getCount = new SqlCommand("Select count(distinct ItemName) from FarmerItems where Category ='" + categoryName + "'", conn);

int count = Convert.ToInt16(getCount.ExecuteScalar().ToString());

productNames = new String[count];

//Declare the sql command

SqlCommand getProductCmd = new SqlCommand("Select distinct ItemName from FarmerItems where Category ='" + categoryName + "'", conn);

SqlDataReader reader = getProductCmd.ExecuteReader();

int i = 0;

while (reader.Read())

{

productNames[i++] = reader[0].ToString();

}

reader.Close();

//close the connection

conn.Close();

return productNames;

}

}

}

Webconfig

<?xml version="1.0"?>

<configuration>

<connectionStrings>

<add name="MercadoDb" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=MercadoDatabase;Integrated Security=SSPI" providerName="System.Data.SqlClient"/>

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.0" />

</system.web>

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior>

<!-- To avoid disclosing metadata information, set the value below to false and remove the metadata endpoint above before deployment -->

<serviceMetadata httpGetEnabled="true"/>

<!-- To receive exception details in faults for debugging purposes, set the value below to true. Set to false before deployment to avoid disclosing exception information -->

<serviceDebug includeExceptionDetailInFaults="false"/>

</behavior>

</serviceBehaviors>

</behaviors>

<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />

</system.serviceModel>

<system.webServer>

<modules runAllManagedModulesForAllRequests="true"/>

</system.webServer>

</configuration>

FarmerReview Service:

Iservice1.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.ServiceModel.Web;

using System.Text;

namespace FarmerReview

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both code and config file together.

[ServiceContract]

public interface IService1

{

[OperationContract]

String PutFarmerReview(String cName, String fName, String rating, String review);

[OperationContract]

FarmerReviewList GetFarmerReview(String fName);

[OperationContract]

String[] GetFarmers();

}

[DataContract]

public class FarmerReview

{

[DataMember]

public String fName { get; set; }

[DataMember]

public String cName { get; set; }

[DataMember]

public int rating { get; set; }

[DataMember]

public String review { get; set; }

}

[DataContract]

public class FarmerReviewList

{

[DataMember]

public List<FarmerReview> reviewList { get; set; }

}

}

Iservice1.svc.cs

/\*

\* This Rest Service is used to Store the Farmer Reviews and Get the all the Farmer Reviews

\*

\* Author Name: Bharat Viswanadham

\*

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.ServiceModel.Web;

using System.Text;

using System.Data.SqlClient;

using System.Configuration;

namespace FarmerReview

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.

public class Service1 : IService1

{

/\*

\* This Rest Service is used to Store the Farmer Reviews

\* Input is customerName, farmerName, rating, review

\*/

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "PutFarmerReview/{cName}/{fName}/{rating}/{review}")]

public String PutFarmerReview(String cName, String fName, String rating, String review)

{

String result = "";

try{

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["MercadoDb"].ConnectionString);

conn.Open();

int rate = Convert.ToInt16(rating);

SqlCommand putFarmerReviewCmd = new SqlCommand("Insert into farmerReview(customerName,farmerName,rating,comments) values ('"

+ cName + "','" + fName + "'," + rate + ",'" + review + "')", conn);

int rowsaffected = putFarmerReviewCmd.ExecuteNonQuery();

if (rowsaffected == 1)

result = "Success";

else

result = "Failure";

putFarmerReviewCmd.Dispose();

conn.Close();

}

catch(Exception e)

{

e.GetBaseException();

Console.WriteLine(e.Message);

}

return result;

/\*

putFarmerReviewCmd = new SqlCommand("select \* from farmerReview", conn);

SqlDataReader reader = putFarmerReviewCmd.ExecuteReader();

String c="", f="", r="", re="";

while (reader.Read())

{

c = reader[0].ToString();

f = reader[1].ToString();

r = reader[2].ToString();

re = reader[3].ToString();

}

String result = c + f + r + re;

return result;

\*/

}

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "GetFarmerReview/{fName}")]

public FarmerReviewList GetFarmerReview(String fName)

{

FarmerReviewList fReviews = new FarmerReviewList();

fReviews.reviewList = new List<FarmerReview>();

try

{

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["MercadoDb"].ConnectionString);

conn.Open();

SqlCommand getReviewcountCmd = new SqlCommand("select distinct count(\*) from farmerReview where farmerName = '" + fName + "'", conn);

int reviewCount = Convert.ToInt16(getReviewcountCmd.ExecuteScalar().ToString());

conn.Close();

FarmerReview[] reviewResults = new FarmerReview[reviewCount];

conn.Open();

SqlCommand getReviewsCmd = new SqlCommand("select distinct \* from farmerReview where farmerName ='" + fName + "'", conn);

SqlDataReader reader = getReviewsCmd.ExecuteReader();

int i = 0;

while (reader.Read())

{

reviewResults[i] = new FarmerReview();

reviewResults[i].cName = reader["customerName"].ToString();

reviewResults[i].fName = reader["farmerName"].ToString();

reviewResults[i].rating = Convert.ToInt32(reader["rating"].ToString());

reviewResults[i].review = reader["comments"].ToString();

fReviews.reviewList.Add(reviewResults[i]);

i++;

}

}

catch(Exception e)

{

e.GetBaseException();

Console.WriteLine(e.Message);

}

return fReviews;

}

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "GetFarmers")]

public String[] GetFarmers()

{

String[] farmerNames ={""};

try{

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["MercadoDb"].ConnectionString);

conn.Open();

SqlCommand getReviewcountCmd = new SqlCommand("select count (username) from farmerregistration", conn);

int reviewCount = Convert.ToInt16(getReviewcountCmd.ExecuteScalar().ToString());

conn.Close();

farmerNames = new String[reviewCount];

conn.Open();

SqlCommand getReviewsCmd = new SqlCommand("select username from farmerregistration", conn);

SqlDataReader reader = getReviewsCmd.ExecuteReader();

int i = 0;

while (reader.Read())

{

farmerNames[i++] = reader[0].ToString();

}

conn.Close();

return farmerNames;

}

catch(Exception e)

{

e.GetBaseException();

Console.WriteLine(e.Message);

}

return farmerNames;

}

}

}

Webconfig

<?xml version="1.0"?>

<configuration>

<connectionStrings>

<add name="MercadoDb" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=MercadoDatabase;Integrated Security=SSPI" providerName="System.Data.SqlClient"/>

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.0" />

</system.web>

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior>

<!-- To avoid disclosing metadata information, set the value below to false and remove the metadata endpoint above before deployment -->

<serviceMetadata httpGetEnabled="true"/>

<!-- To receive exception details in faults for debugging purposes, set the value below to true. Set to false before deployment to avoid disclosing exception information -->

<serviceDebug includeExceptionDetailInFaults="false"/>

</behavior>

</serviceBehaviors>

</behaviors>

<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />

</system.serviceModel>

<system.webServer>

<modules runAllManagedModulesForAllRequests="true"/>

</system.webServer>

</configuration>